SYSTEM FOR SELECTIVELY ENABLING OPERATING MODES OF A DEVICE

BACKGROUND

I. FIELD

[0001] The present invention relates generally to the efficient operation of a device, and more particularly, to a system for selectively enabling operating modes of a device.

II. DESCRIPTION OF THE RELATED ART

[0002] Advances in technology have resulted in the development of a variety of devices for home, office, and personal use. For example, computer systems, such as desktops, notebooks, and tablet computers have become powerful tools for use at home or in office environments. Personal devices, such as wireless telephones, personal digital assistants (PDAs), and paging devices have also become more powerful and are now widely used.

[0003] Many devices now comprise complex hardware and software that run operating systems such as UNIX, LINUX, or similar operating systems. Generally, these operating systems provide for multiple modes of operation. For example, most systems provide privileged and non-privileged modes of operation. Programs executing in the privileged mode of operation are allowed to access memory and system resources without limitation. Programs executing in the non-privileged mode are restricted from accessing certain memory regions and/or device systems. This configuration provides a level of protection to important memory or device functions. For example, by executing third-party applications in the non-privileged mode, important memory regions and device functions can be protected from unauthorized access. Also, such an arrangement allows the system to isolate faults during execution.

[0004] In systems without multiple modes of operation, there need be no concern for protecting the integrity of privileged mode, and existing software is not structured in a way to accommodate a proper division between privileged and non-privileged operation. Software written in this manner is incompatible within a system that supports privileged and non-privileged modes of operation.

[0005] These compatibility issues complicate the task of moving a large body of software from a system that does not support multiple modes of execution to a system

that does. The software as a whole cannot be tested intact until all the pieces have been migrated to the multi-mode model.

[0006] Therefore, what is needed is a system that supports multiple modes of operation and can optionally run software in one mode of operation without change to the software. In such a system, applications can be executed and tested with older software while they are being converted to support multiple modes of operation.

SUMMARY

[0007] In one or more embodiments, a system comprising methods and/or apparatus is provided that allows operating modes of a device to be selectively enabled. For example, in one embodiment, a mechanism is provided that allows the selection of operating modes to occur during a device initialization process, and as a result, the operating mode selection does not have to be compiled into the executable program that runs on the device.

[0008] In one or more embodiments, a non-volatile flag is used to determine the selected operating modes of the device. For example, the flag resides in non-volatile memory and may only be set or cleared by the device manufacturer or other authorized personnel. During device initialization, the flag is tested to determine the selected operating mode of the device.

[0009] In one embodiment, a method is provided for selectively enabling operating modes of a device during a device initialization, wherein the operating modes comprise a privileged mode and a non-privileged mode. The method comprises determining during the device initialization whether the device is to operate in the privileged mode or in both the privileged and non-privileged modes, and enabling the privileged mode if it is determined that the device is to operate only in the privileged mode. The method also comprises enabling both the privileged and the non-privileged modes if it is determined that the device is to operate in both the privileged and the non-privileged modes.

[0010] In another embodiment, apparatus is provided for selectively enabling operating modes of a device during a device initialization, wherein the operating modes comprise a privileged mode and a non-privileged mode. The apparatus comprises a flag and selection logic that operates to read the flag to set the operating mode of the device, wherein if the flag is set the selection logic enables the privileged mode, and if the flag is not set, the selection logic enables both the privileged and non-privileged modes.

[0011] In another embodiment, apparatus is provided for selectively enabling operating modes of a device during a device initialization, wherein the operating modes comprise a privileged mode and a non-privileged mode. The apparatus comprises means for determining during the device initialization whether the device is to operate in the privileged mode or in both the privileged and non-privileged modes. The apparatus also comprises means for enabling only the privileged mode if it is determined that the device is to operate only in the privileged mode. The apparatus also comprises means for enabling both the privileged and the non-privileged modes if it is determined that the device is to operate in both the privileged and the non-privileged modes.

[0012] In another embodiment, a computer-readable media is provided comprising instructions, which when executed by a processor in a device, operate to selectively enable operating modes of a device during a device initialization, wherein the operating modes comprise a privileged mode and a non-privileged mode. The computer-readable media comprises instructions for determining during the device initialization whether the device is to operate in the privileged mode or in both the privileged and non-privileged modes. The computer-readable media also comprises instructions for enabling only the privileged mode if it is determined that the device is to operate only in the privileged mode. The computer-readable media also comprises instructions for enabling both the privileged and the non-privileged modes if it is determined that the device is to operate in both the privileged and the non-privileged modes.

[0013] Other aspects, advantages, and features of the present invention will become apparent after review of the hereinafter set forth Brief Description of the Drawings, Detailed Description of the Invention, and the Claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The foregoing aspects and the attendant advantages of the embodiments described herein will become more readily apparent by reference to the following detailed description when taken in conjunction with the accompanying drawings wherein:

[0015] **FIG. 1** shows a device that includes one embodiment of an operating mode selection system;

[0016] **FIG. 2** shows a functional block diagram of a device comprising one embodiment of an operating mode selection system; and

[0017] **FIG. 3** shows a flow diagram that illustrates the operation of one embodiment of an operating mode selection system for use in a device.

DETAILED DESCRIPTION

[0018] The following detailed description describes an operating mode selection system that allows operating modes of a device to be selectively enabled. For example, the system operates to provide

[0019] In one or more embodiments, the selection system interacts with a runtime environment (or operating system) executing on the device that is used to simplify operation of the device, such as by providing generalized calls for device specific resources. One such runtime environment is the Binary Runtime Environment for Wireless™ (BREW™) software platform developed by QUALCOMM, Inc., of San Diego, California. The following description describes a device executing a runtime environment, such as the BREW software platform. However, in one or more embodiments, the registration system is suitable for use with other types of runtime environments to provide fast and efficient registration of privileged function hooks in a variety of devices, including generating systems or other controlling or monitoring programs. For example, the devices may include, but are not limited to, desktop computers, notebook computers, handheld computers, and portable devices, such as wireless telephones, pagers, PDAs, email devices, tablet computers, or other type of computing devices.

[0020] **FIG. 1** shows one embodiment of an operating mode selection system **100** that operates to allow operating modes of a device **102** to be selectively enabled. The system **100** may be part of a home computer, office computer, or personal device, such as a wireless telephone or PDA, or any other type of computing device.

[0021] In one embodiment, the device **102** includes a runtime environment **104** (i.e., BREW) that provides at least two modes of operation; namely a non-privileged mode (NP) and a privileged mode (P). The non-privilege mode of operation is used to restrict the access of applications running on the device and to provide fault detection. For example, applications running in non-privileged mode are restricted to selected memory regions and may be denied access to device registers or other resources. The privileged mode of operation allows applications running under this mode to access memory or device resources without limitation.

[0022] During system initialization, the device begins operating in the privileged mode **106**. In one embodiment, selection logic **108** executes to selectively enable operating modes of the device **102**. The selection logic **108** determines if the device **102** is to operate in the privileged mode only, as shown at **110**, or if the device **102** is to operate with both privileged and non-privileged modes, as shown by **112** and **114**, respectively.

[0023] Because the selection logic **108** operates during system initialization, it is possible to easily change the operating mode configuration of the device without having to compile a new executable. Thus, the selection system is well suit for use during system development and testing because operating mode selection can be performed through a simple mechanism that is fast and easy to use.

[0024] **FIG. 2** shows a functional block diagram of the device **102** comprising one embodiment of an operating mode selection system. The device **102** comprises processing logic **202**, device resources **206**, and a non-volatile memory **234** that are coupled to an internal data bus **204**. The device resources **206** comprise hardware, software, memory, logic, and/or other resources that allow the device to interface to various internal and external resources, devices, or systems. Also coupled to the processing logic **202** are code memory **208** and data memory **210**.

[0025] In one or more embodiments, the processing logic **202** comprises a CPU, processor, gate array, hardware logic, memory elements, virtual machine, software, I/O interfaces, and/or any combination of hardware and software. Thus, the processing logic **202** generally comprises logic to execute machine-readable instructions. For example, instructions may be loaded into the device **102** from a computer-readable media, such as a floppy disk, CDROM, Flash memory, or other computer-readable media that interfaces to the device **102** via the device resources **206**. In another embodiment, the instructions may be downloaded into the device **102** from a network resource, such as a network server or any other type of network resource that interfaces to the device **102** via the device resources **206**. The instructions, when executed by the processing logic **202**, provide one or more embodiments of an operating mode selection system as described herein.

[0026] In one embodiment, code memory **208** comprises RAM, ROM, FLASH, EEROM, or any other suitable type of memory, or a combination thereof. A first memory management unit (MMU) **216** is coupled to the processing logic **202** and the code memory **208**. The MMU **216** operates to partition the code memory **206** into any combination of a privileged region (P) **212** and a non-privileged region (NP) **214**. For

example, the MMU **216** may partition the code memory into a single privileged partition **218**, or into both non-privileged **214** and privileged **212** partitions. When the non-privileged partition **214** is used, the MMU **216** operates to restrict the operation of non-privileged program code so that the non-privileged code has limited access rights to selected regions of the code memory **208**.

[0027] In one embodiment, the data memory **210** comprises RAM, ROM, FLASH, EEROM, or any other suitable type of memory, or a combination thereof. A second memory management unit (MMU) **220** is coupled to the processing logic **202** and the data memory **210**. The MMU **220** operates to partition the data memory **206** into any combination of a privileged region (P) **222** and a non-privileged region (NP) **224**. For example, the MMU **220** may partition the data memory into a single privileged partition **226**, or into both non-privileged **224** and privileged **222** partitions. When the non-privileged partition **224** is used, the MMU **220** operates to restrict the operation of non-privileged program code so that the non-privileged code has limited access rights to selected regions of the data memory **210**.

[0028] The device **102** also includes selection logic **228** that, in one embodiment, is part of the processing logic **202**. For example, the selection logic **228** may comprise the processing logic **202** executing selected program instructions stored in the code memory **208**. The selection logic **228** operates to selectively control the MMUs **216** and **220**. For example, the selection logic **228** controls the operation of the MMU **216** via the control line **230**. The selection logic **228** also controls the operation of the MMU **220** via the control line **232**. Therefore, operation of the selection logic **228** can be used to selectively enable operating modes of the device **102**.

[0029] The non-volatile memory **234** is used to store a flag that indicates the selected operating modes for the device. In one embodiment, the flag is protected so that only the device manufacturer or other authorized personnel can set or clear the flag. For example, a system integrator may access the memory **234** to set or clear the flag by using a system integration tool.

[0030] During initialization, the selection logic **228** accesses the memory **234** to read the flag. The selection logic **228** then uses the flag to enable the operating mode of the device **102**. For example, in one embodiment, the flag is set high, and the selection system enables only the privileged mode of operation. This is accomplished by the selection logic **228** controlling both the MMUs **216** and **220**. In another embodiment, the flag is set low, and the selection system enables both privileged and non-privileged

modes. This is also accomplished by the selection logic **228** controlling both the MMUs **216** and **220**.

[0031] In one embodiment, the described mode selection system comprises program instructions stored on a computer-readable media, which when executed by the processing logic **202**, provides the functions described herein. In one or more embodiments, the computer-readable media comprises a floppy disk, CD, memory card, FLASH memory device, RAM, ROM, or any other type of memory device.

[0032] **FIG. 3** shows a flow diagram **300** that illustrates the operation of one embodiment of an operating mode selection system for use in a device, for example the device **102**. For clarity, the flow diagram **300** will be described with reference to the device **102** shown in **FIG. 2**. It will be assumed that the device is executing a BREW runtime environment and that the processing logic **202** executes program instructions stored in the code memory **208** to perform the functions described below.

[0033] During operation, an initialization thread of execution **302** is created that represents the initialization of the device **102**.

[0034] At block **304**, a test is performed to determine what operating modes are to be enabled on the device. For example, in one embodiment, the test comprises the selection logic **228** reading the flag in the memory **234** to determine the selected operation mode. If the selection flag is set high, the method proceeds to block **306**. If the selection flag is not set high, the method proceeds to block **308**.

[0035] At block **306**, the operating mode is selected so that the device operates in only the privileged mode. To enable only the privileged mode, the selection logic **228** controls the MMUs **216** and **220**, via the control lines **230** and **232**, respectively.

[0036] At block **308**, the operating mode is selected so that the device operates in both the privileged and non-privileged operating mode. To enable both modes, the selection logic **228** controls the MMUs **216** and **220**, via the control lines **230** and **232**, respectively. After the mode of the device is selected, the method continues the initialization thread of execution **302**.

[0037] It should be noted that the program flow diagram **300** illustrates just one embodiment and that changes, additions, or rearrangements of the flow elements may be made without deviating from the scope of the invention.

[0038] Accordingly, while one or more embodiments of methods and/or apparatus for an operating mode selection system have been illustrated and described herein, it will be appreciated that various changes can be made to the embodiments without departing

from their spirit or essential characteristics. Therefore, the disclosures and descriptions herein are intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

I CLAIM: